Cooperative Recovery of Distributed Storage Systems from Multiple Losses with Network Coding

Yuchong Hu, Yinlong Xu, Xiaozhao Wang, Cheng Zhan and Pei Li

***Abstract*—This paper studies the recovery from multiple node failures in distributed storage systems. We design a mutually cooperative recovery (MCR) mechanism for multiple node failures. Via a cut-based analysis of the information flow graph, we obtain a lower bound of maintenance bandwidth based on MCR. For MCR, we also propose a transmission scheme and design a linear network coding scheme based on (n, k) strong-MDS code, which is a generalization of (n, k) MDS code. We prove that the maintenance bandwidth based on our transmission and coding schemes matches the lower bound, so the lower bound is tight and the transmission scheme and coding scheme for MCR are optimal. We also give numerical comparisons of MCR with other redundancy recovery mechanisms in storage cost and maintenance bandwidth to show the advantage of MCR.**

***Index Terms*—Network coding, Distributed storage system, Erasure coding, Multiple loss recovery**

I. Introduction

DISTRIBUTED storage systems have been becoming increasingly popular with the rapid growth of storage volume, bandwidth, and computational resources. Some recently built distributed storage systems are OceanStore[5], PAST[1], CFS[2], Total Recall[3], Farsite[4], etc. An important goal for such systems is to keep available whenever any nodes (or disks) fail, which needs to maintain a certain level of redundancy. Redundancy can be achieved by two common mechanisms, *replication* and *erasure coding*. Replication is the simplest redundancy scheme, where c identical copies of a file are kept at c nodes, each node with one copy. The other is an (n, k) Maximum-Distance Separable (MDS) code, where each file of size M bytes is divided into k fragments of size M/k bytes and the k fragments are encoded into n fragments stored at n nodes, each node with one fragment, where $n > k$. The key property of erasure coding is that the original file can be reconstructed from any k fragments. Compared with replication, erasure coding uses an order of magnitude less bandwidth and storage to provide the same system availability[12].

With erasure coding in distributed storage system, it is not trivial to create new fragments to recover from node failures. There are two alternative approaches [6] to create a new fragment. One is that a new node (which will store the new fragment) will download k fragments and reconstruct the original file, from which a new fragment is produced, such as IDA [7]. This is very costly because it will download the entire file to create a fragment, with a network traffic of M bytes. The other is to maintain an extra full copy of the file at one node, along with the fragments at the rest of nodes. Thus, the new node just needs to download one fragment from the node maintaining a full copy, with a network traffic of M/k bytes. Though the second approach reduces the cost of creating a fragment, it is an asymmetric strategy maintaining two types of redundancy, where the task of redundancy maintenance becomes more complex. Therefore the benefits of asymmetric erasure coding are so limited in some cases that they can easily be outweighed by the extra complexity of the asymmetric redundancy mechanism[6].

Dimakis et al.[8][10] prove that the file reconstruction problem in distributed storage systems is equivalent to the multicasting problem. They also propose two symmetric mechanisms for maintaining redundancy using erasure coding to reduce the recovery bandwidth overhead (*we call it maintenance bandwidth in this paper*). One is minimum-storage regenerating (MSR) codes, an (n, k) MDS code that can be efficiently repaired. In the special case of the $(n, k) = (14, 7)$, a new node only needs to download $0.265M$ bytes to repair a new fragment. The other is minimum-bandwidth regenerating (MBR) codes, which uses larger fragments than MSR, but is with lower maintenance bandwidth. As the same example of $(n, k) = (14, 7)$, a new node needs to download only $0.1857M$ bytes to repair a new fragment, 29.9% less than MSR. Yunnan Wu et al.[9] give techniques for constructing MSR and MBR codes that achieve the optimal tradeoffs between storage efficiency and maintenance bandwidth.

To the best of our knowledge, the above redundancy recovery mechanisms are designed for one node failure. However, the situation of recovery from multiple losses often occurs in distributed storage systems. For example, some systems like Total Recall [3] reconstruct fragments with lazy repair policy, where a recovery is triggered only when the total amount of losses reaches a given threshold. Besides lazy repair policy, there are many other situations where multiple nodes fail at the same time in real storage systems, such as churn, i.e., a large percent of nodes often join and/or leave the network simultaneously. Moreover, some peer-to-peer systems have to

Manuscript received 12 May 2009; revised 4 September 2009. This work is supported by the National Science Foundation of China under Grant No. 60773036.

The authors are with the School of Computer Science & Technology, University of Science & Technology of China and Key Laboratory on High Performance Computing, Anhui Province, e-mail:{churhu,xiazhiray,zhanchen,lipei}@mail.ustc.edu.cn; ylxu@ustc.edu.cn.

Digital Object Identifier 10.1109/JSAC.2010.100216.

face with the sudden disconnections of multiple nodes because the power is cut off over a wide area, or a large number of maliciously controlled peers leave the network at the same time.

Although most of the existing recovery mechanisms can be used for repairing multiple node failures one by one, they may not be optimal. Take MSR and MBR as examples. Suppose that there exist n storage nodes initially, and then r nodes fail simultaneously. It is necessary to recover the lost data at r new nodes. The repaired data of each new node only comes from $n-r$ available nodes based on MSR and MBR recovery mechanisms, and the bandwidth resources between r new nodes are not used. So it is natural that we consider the problem how much bandwidth resources between new nodes can be exploited to reduce the maintenance bandwidth overhead if the new nodes are repaired cooperatively.

In this paper we will present a mutually cooperative recovery (named MCR) mechanism for multi-loss recovery. Different from MSR and MBR where all the new nodes repair the lost data one by one, in MCR all the new nodes repair the lost data cooperatively and simultaneously. Each new node in MSR and MBR connects to only $n-r$ available nodes for recovery while each new node in MCR chooses $n-1$ nodes (both all the $n-r$ survival nodes and the other $r-1$ new nodes) for recovery. Because the lost data of each new node in MCR can be retrieved from more nodes than MSR and MBR, the number of linearly dependent packets transmitted for recovery will be less and the recovery efficiency will be higher. Thus, MCR may perform better than MSR and MBR in terms of maintenance bandwidth.

We will propose a model for the multi-loss recovery in distributed storage systems in MCR based on the information flow graph, and analyze the minimal maintenance bandwidth with the maxflow-mincut theorem. We will show that the MCR mechanism is more efficient than MSR and MBR by numerical comparisons. We will also present a MCR transmission scheme and introduce a *strong-*$MDS\ Property$ which is a generalization of $MDS\ Property$ [9] for the proof of the correctness of MCR. At last, we will prove that there exists a random linear coding with the minimal maintenance bandwidth in MCR while keeping the *strong-*$MDS\ Property$.

After detailed definitions and notations in Section II, in Section III we will analyze the lower bound of maintenance bandwidth for multi-loss recovery in MCR, and will give a comparison among MCR, MSR and MBR. In Section IV, we will elaborate the MCR transmission scheme and present a *strong-*MDS random linear coding scheme. Section V concludes this paper.

II. Problem Statement

Assume that there are a large number of storage nodes characterized to have an identical storage capability and the communications between any two nodes are symmetric in a distributed storage system. The original file is (n,k) MDS encoded and the n encoded fragments are stored evenly at n nodes chosen from the system. When r nodes become unavailable, the system chooses another r nodes to repair the lost fragments to keep the same level of redundancy. See Fig. 1 for an illustration of distributed storage networks with multiple node failure recovery. This paper studies the maintenance bandwidth for multi-loss recovery and designs an efficient recovery mechanism.

The problem of recovery from multiple losses is stated as follows.

- At the beginning, the fragments at all the storage nodes are created from the data distribution of the source node. But as time goes by, the source node may leave the storage network. In order to facilitate the flow analysis, we define a *virtual source* (VS) which always connects to the initial storage nodes.
- Assume that an initial set of n nodes $X_1,\ldots,X_n$ are chosen to store the original file. Each node stores one encoded fragment. The destination node D can connect to any k nodes out of $X_1,\ldots,X_n$ to download k fragments for the file reconstruction.
- When r storage nodes (say $X_{n-r+1},\ldots,X_n$) become unavailable, in order to maintain the same level of redundancy, the system will select another r new nodes (denoted as $Y_1,\ldots,Y_r$) to repair the lost fragments. Each of the new nodes stores one encoded fragment which is created by the downloaded data from other available nodes. After the recovery, n nodes $X_1,\ldots,X_{n-r}$ (named "old nodes"), $Y_1,\ldots,Y_r$ (nameds "new nodes") store the original file. The destination node D can connect to any k nodes out of $X_1,\ldots,X_{n-r},Y_1,\ldots,Y_r$ to download k fragments for the file reconstruction.

We denote the above distributed storage network with multi-loss recovery as $DSN(n,k,r)$. In the following of this paper, we will obtain a lower bound of maintenance bandwidth in $DSN(n,k,r)$ based on MCR, and propose MCR transmission and coding schemes which match the lower bound.

III. Mutually Cooperative Recovery(MCR)

Our study bases on the assumption that all the new nodes can mutually cooperatively complete their fragment reconstructions. In this section, we will firstly introduce the system model based on MCR. And then we will use the information flow theory to analyze the lower bound of maintenance bandwidth of MCR. Finally, we will compare MCR, MSR and MBR in maintenance bandwidth and storage consumption, and conclude that MCR outperforms others.

A. Model based on MCR

The repair process of our mutually cooperative recovery in $DSN(n,k,r)$ is specified as follows.

(1) Each new node Y_j $(1\le j\le r)$ firstly connects to all the existing available nodes $X_1,\ldots,X_{n-r}$ to download encoded data $\beta_{1,j},\ldots,\beta_{n-r,j}$, where $\beta_{i,j}$ is the data transmitted from X_i to Y_j $(1\le i\le n-r)$.

(2) And then each new node Y_j connects to all the other new nodes $Y_1,\ldots,Y_{j-1},Y_{j+1},\ldots,Y_r$ to download encoded data $\beta'_{1,j},\ldots,\beta'_{j-1,j},\beta'_{j+1,j},\ldots,\beta'_{r,j}$, where $\beta'_{j',j}$ $(j'\neq j$ and $1\le j'\le r)$ is the data transmitted from $Y_{j'}$ to Y_j.

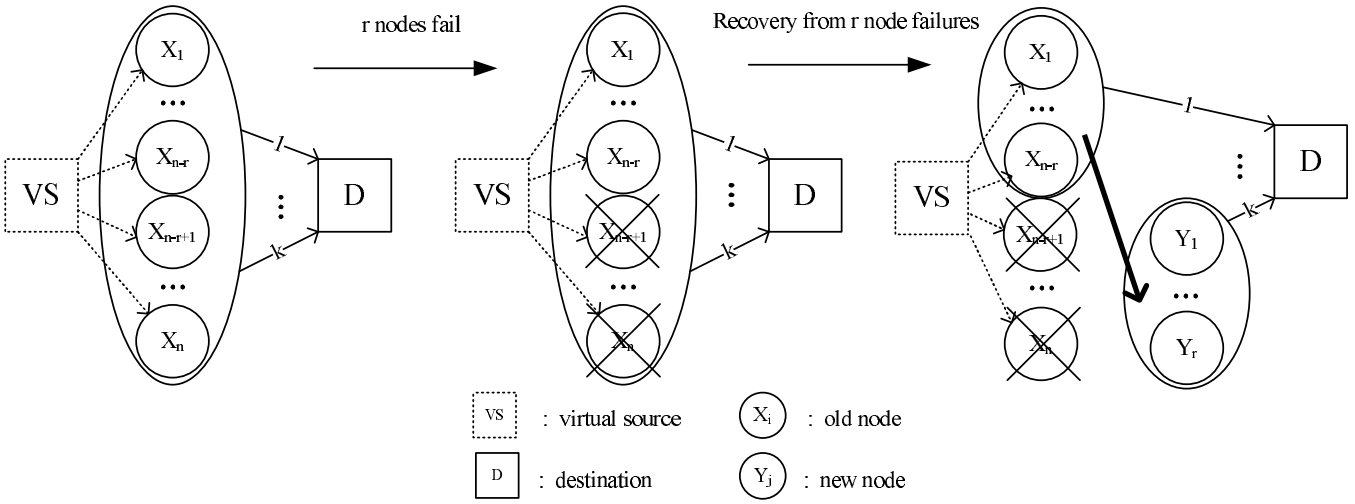


Fig. 1. An illustration of $DSN(n,k,r)$

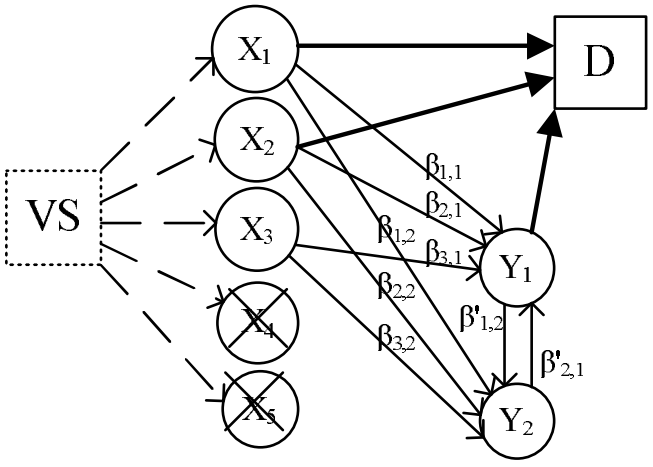


Fig. 2. An illustration of $DSN(n,k,r)$ in MCR, where $n=5, k=3, r=2$

(3) Finally, each new node Y_j generates an encoded fragment based on all the downloaded data $\beta_{1,j},\ldots,\beta_{n-r,j}$ and $\beta'_{1,j},\ldots\beta'_{j-1,j},\beta'_{j+1,j},\ldots,\beta'_{r,j}$.

See Fig. 2 for an illustration of $DSN(n,k,r)$ in MCR.

B. Assumptions in MCR

From a practical view, a distributed storage network will be very difficult to be implemented and managed if $\beta_{1,j},\ldots,\beta_{n-r,j}$ and $\beta'_{1,j},\ldots\beta'_{j-1,j},\beta'_{j+1,j},\ldots,\beta'_{r,j}$ are not equal. So we assume that $\beta_{i,j}$ and $\beta'_{j',j}$ is the same as β. And thus the maintenance bandwidth (the total bandwidth overhead for the recovery) is

$$\begin{aligned}\gamma_{MCR} &= \sum_{1\le i\le n-r, 1\le j\le r}\beta_{i,j}+\sum_{1\le j',j\le r, j'\ne j}\beta'_{j',j}\\ &= (n-r)r\beta+(r-1)r\beta\\ &= (n-1)r\beta\end{aligned}$$

Our goal is to analyze the bandwidth overhead γ_{MCR} while keeping (n,k) MDS property, i.e., the destination D can always reconstruct the original file by connecting to any k out of n nodes. In Subsection D, we will give a lower bound of γ_{MCR}, and in Section IV we will present transmission and coding schemes to realize MCR with γ_{MCR} matching the lower bound.

C. Information flow graph $G(n,k,r,\beta)$

In this subsection, we will formulate the multi-loss recovery in $DSN(n,k,r)$ in MCR as an information flow graph $G(n,k,r,\beta)$, a similar idea in [8].

The information flow graph $G(n,k,r,\beta)$ corresponding to $DSN(n,k,r)$ is constructed as follows.

1) Nodes in $G(n,k,r,\beta)$:

- Each old node X_i in $DSN(n,k,r)$ is represented by an input end X_i^{in}, an output end X_i^{out} and a directed edge $X_i^{in}\to X_i^{out}$ with a capacity of M/k which equals to the amount of data stored at node X_i, the same as in [8].
- Each new node Y_j in $DSN(n,k,r)$ will transmit data to each of the other new nodes for the recovery. In order to avoid directed cycles of the information flow graph, a middle end Y_j^{mid} and a directed edge $Y_j^{in}\to Y_j^{mid}$ with infinite capacity are added for Y_j. The directed edge $Y_j^{mid}\to Y_j^{out}$ is with a capacity of M/k which equals to the amount of data stored at node Y_j.

2) Edges in $G(n,k,r,\beta)$:

- The edge from VS to each of the initial nodes in $DSN(n,k,r)$ is represented by a directed edge of infinite capacity in $G(n,k,r,\beta)$.
- The edge from an old node X_i to a new node Y_j in $DSN(n,k,r)$ is represented by a directed edge from X_i^{out} to Y_j^{in} with a capacity of β in $G(n,k,r,\beta)$.
- The edge from a new node Y_j to another new node $Y_{j'}$ in $DSN(n,k,r)$ is represented by a directed edge from Y_j^{in} to $Y_{j'}^{mid}$ with a capacity of β in $G(n,k,r,\beta)$.
- To reconstruct the original file, D will connect to k new nodes and/or old nodes. Each of the edges from the selected k nodes to D in $DSN(n,k,r)$ can be represented by a directed edge of infinite capacity in $G(n,k,r,\beta)$.

See Fig. 3 for an illustration of $G(n,k,r,\beta)$.

D. Lower bound of maintenance bandwidth

In this subsection, we will find the lower bound of β by studying the capacity of the min-cut of $G(n,k,r,\beta)$.

Our goal is to give a lower bound of β to ensure that $DSN(n,k,r)$ keeps the (n,k) MDS property, i.e., the destination D can always reconstruct the original file by connecting to any k out of n nodes. Different connection choices of D will introduce different information flow graphs and correspondingly different capacities of min-cuts. To keep the (n,k) MDS property in $DSN(n,k,r)$, each of the capacities of min-cuts in all the possible information flow graphs must be more than or equal to the original file size M bytes. Otherwise, the point-to-point maxflow from VS to D is less than M bytes and thus D cannot obtain enough data to reconstruct the original file. So the following Lemma 1 concludes.

Lemma 1: To keep (n,k) *MDS* property in $DSN(n,k,r)$, each of the capacities of min-cuts separating VS from D in all possible information flow graphs $G(n,k,r,\beta)$ must be not smaller than M bytes.

By Lemma 1, it is necessary that each min-cut of $G(n,k,r,\beta)$ is not smaller than M bytes, which will be used to prove the following Lemma 2.

Lemma 2: Let $(S,\bar{S})$ be the cut of $G(n,k,r,\beta)$, where $VS \in S$, $D \in \bar{S}$. Suppose that each new node firstly downloads β bytes from each of $n-r$ old nodes and then downloads β bytes from each of the other $r-1$ new nodes, and D can connect to any k out of n nodes (including $n-r$ old nodes and r new nodes) for the file reconstruction. If $\beta \geq M/[k(n-k)]$, each of the capacities of min-cuts of all possible $G(n,k,r,\beta)$ is not smaller than M bytes.

Proof: Assume that D connects to t new nodes $Y_1,\ldots,Y_t$ and $k-t$ old nodes $X_1,\ldots,X_{k-t}$ for the file reconstruction. We figure out the capacities of the possible min-cuts $c(S,\bar{S})$ of $G(n,k,r,\beta)$, where $VS \in S, D \in \bar{S}$.

Since the capacity of the edge from VS to X_i^{in} is infinite, X_i^{in} will not be in $\bar{S}$. For the same reason, Y_j^{out} will not be in S. Therefore, the nodes in $\bar{S}$ can be classified into three cases:

- *Case 1*: X_i^{out} is in $\bar{S}$
- *Case 2*: Y_j^{out}, Y_j^{in} and Y_j^{mid} are all in $\bar{S}$
- *Case 3*: Y_j^{out} is in $\bar{S}$, but Y_j^{in} and Y_j^{mid} are in S.

Let t_c be the number of corresponding X_i in Case 1, r_c be the number of corresponding Y_j in Case 2, and w_c be the number of corresponding Y_j in Case 3. See Fig. 3 for an illustration of a cut in $G(n,k,r,\beta)$ based on MCR.

Since the capacity of each of the edges from $X_1^{out},\ldots,X_{k-t}^{out}$ to D is infinite,

$$t_c \geq k-t. \tag{1}$$

For each node in $\bar{S}$, there always exists a path from VS to D passing through it, so

$$r_c + w_c \geq t. \tag{2}$$

Now we calculate the capacity of a cut in $G(n,k,r,\beta)$ by the following four parts:

- *Part 1*: The sum of capacities of edges from X_i^{in} to X_i^{out} for X_i^{in} in S and X_i^{out} in $\bar{S}$ is $(M/k)t_c$.

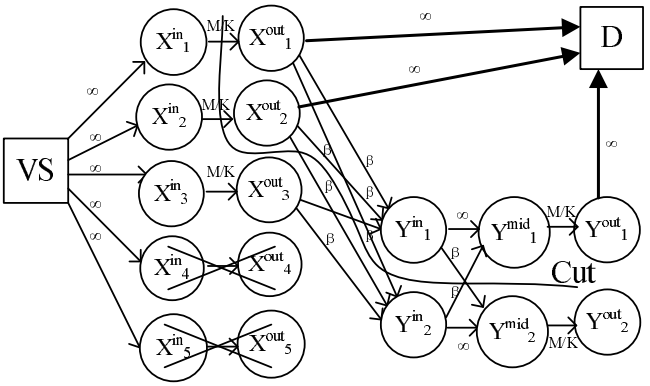


Fig. 3. An illustration of a cut passing through information flow graph $G(5,3,2,\beta)$ in MCR , where $n=5,k=3,r=2,t=1,t_c=2,r_c=1,w_c=0$.

- *Part 2*: The sum of capacities of edges from X_i^{out} to Y_j^{in} for X_i^{out} in S and $Y_j^{out}, Y_j^{in}, Y_j^{mid}$ in $\bar{S}$ is $\beta(n-r-t_c)r_c$.
- *Part 3*: The sum of capacities of edges between Y_j^{in} and $Y_{j'}^{mid}$ for Y_j^{in} in S and $Y_{j'}^{mid}$ in $\bar{S}$ is $\beta(r-r_c)r_c$.
- *Part 4*: The sum of capacities of edges from Y_j^{mid} to Y_j^{out} for Y_j^{mid} in S and Y_j^{out} in $\bar{S}$ is $(M/k)w_c$.

So the capacity of $(S,\bar{S})$ in $G(n,k,r,\beta)$ is

$$\begin{aligned} c(S,\bar{S}) &= (M/k)t_c + \beta(n-r-t_c)r_c \\ &\quad +\beta(r-r_c)r_c + (M/k)w_c \\ &= (M/k-\beta r_c)t_c + \beta(n-r)r_c \\ &\quad +\beta(r-r_c)r_c + (M/k)w_c. \end{aligned} \tag{3}$$

We analyze $c(S,\bar{S})$ for two cases of $(M/k-\beta r_c)$.

- *Case 1*:

$$M/k - \beta r_c \geq 0. \tag{4}$$

By (1) and (4), (3) implies

$$\begin{aligned} c(S,\bar{S}) &\geq (M/k-\beta r_c)(k-t) + \beta(n-r)r_c \\ &\quad +\beta(r-r_c)r_c + (M/k)w_c. \end{aligned} \tag{5}$$

By (2), (5) implies

$$\begin{aligned} c(S,\bar{S}) &\geq (M/k-\beta r_c)(k-t) + \beta(n-r)r_c \\ &\quad +\beta(r-r_c)r_c + (M/k)(t-r_c) \\ &= M-(M/k)r_c + \beta[nr_c - r_c(k-t) - r_c^2]. \end{aligned}$$

Because $\beta \geq M/[k(n-k)]$,

$$\begin{aligned} c(S,\bar{S}) &\geq M-(M/k)r_c \\ &\quad +M[nr_c - r_c(k-t) - r_c^2]/[k(n-k)] \\ &= M + Mr_c(t-r_c)/[k(n-k)]. \end{aligned}$$

And because $t \geq r_c \geq 0, c(S,\bar{S}) \geq M$.

When $t_c = k-t, r_c+w_c = t, t = r_c$ and $\beta = M/[k(n-k)], c(S,\bar{S}) = M$. It means that there are some cases where the capacity of the min-cut is equal to M for $\beta = M/[k(n-k)]$. By Lemma 1, $\beta = M/[k(n-k)]$ is a lower bound.

- *Case 2*:

$$M/k - \beta r_c < 0. \tag{6}$$

Because there are $n-r$ old nodes being available,

$$t_c \leq n-r. \tag{7}$$

By (6) and (7), (3) implies

$$\begin{aligned} c(S,\bar{S}) &\geq (M/k-\beta r_c)(n-r)+\beta(n-r)r_c \\ &\quad +\beta(r-r_c)r_c+(M/k)w_c \\ &= (M/k)(n-r)+\beta(r-r_c)r_c \\ &\quad +(M/k)w_c. \end{aligned} \tag{8}$$

Because any k out of $n-r$ old nodes can be connected to D for the file reconstruction,

$$n-r \geq k. \tag{9}$$

By (9) and $r \geq r_c$, (8) implies

$$c(S,\bar{S}) \geq M.$$

By Case 1 and Case 2, Lemma 2 concludes. ∎

Lemma 3: [8] If each of the capacities of min-cuts of all possible $G(n,k,r,\beta)$ is not smaller than the original file size M, there exists a random linear network coding scheme guaranteeing that D can reconstruct the original file for any connection choice(i.e., (n,k) MDS property), with a probability that can be driven arbitrarily to 1 by increasing the field size of F.

Lemma 2, and 3 allow us to provide a complete characterization of the maintenance bandwidth cost associated with the existence of a linear network coding which keeps the (n,k) MDS property.

Theorem 1: Given a distributed storage network $DSN(n,k,r)$, and an original file of size M bytes, there exists a coding scheme such that the (n,k) MDS property is still kept after the recovery if β is not smaller than $M/[(n-k)k]$.

Proof: By Lemma 2, if β is not smaller than $M/[(n-k)k]$, each of the capacities of min-cuts of all possible $G(n,k,r,\beta)$ is not smaller than M. By Lemma 3, if each of the capacities of min-cuts of all possible $G(n,k,r,\beta)$ is not smaller than M, there exists a linear network coding which maintains (n,k) MDS property. So Theorem 1 concludes. ∎

In Section IV, we will propose a transmission scheme and a deterministic construction of coding in MCR while β is set as $M/[(n-k)k]$. So the lower bound in Lemma 2 is tight, and the transmission and coding schemes proposed in Section IV are optimal.

E. Comparisons of MCR, MSR and MBR

In this subsection, we will compare MCR with other existing symmetric redundancy recovery mechanisms using erasure coding, and show that MCR costs lower storage and maintenance bandwidth while keeping the same redundancy level from multi-loss recovery.

Consider the following distributed storage scenario. There are n initial nodes which store an original file of size M. Each node stores α bytes. After some time, r storage nodes become unavailable. Then the multi-loss recovery is triggered. Each of r new node downloads β bytes from each of any d available nodes, and therefore the total maintenance bandwidth $\gamma = d\beta$ bytes . We further restrict our attention to the symmetric setup where the original file can be retrieved from any k nodes. Under the above scenario, we will give the storage costs and maintenance bandwidths of different symmetric redundancy recovery mechanisms in the following.

TABLE I
PERFORMANCE COMPARISON OF IEC, MCR, MSR AND MBR

	Total node storage cost	Maintenance bandwidth
IEC	$M*R_l$	$M*(R_l-R_s)$
MCR	$M*R_l$	$M*(R_l-R_s)*\frac{kR_l-1}{kR_l-k}$
MSR	$M*R_l$	$M*(R_l-R_s)*\frac{kR_s}{kR_s-k+1}$
MBR	$M*\frac{2kR_s}{2kR_s-k+1}*R_l$	$M*(R_l-R_s)*\frac{2kR_s}{2kR_s-k+1}$

IEC: We take ideal erasure coding for comparisons. With IEC, a new node needs to store M/k and consume an ideal traffic of M/k to repair the data. So the storage cost is $(M/k)\cdot n$, and the maintenance bandwidth is $(M/k)\cdot r$.

MCR: From the above analysis in this paper, when using MCR with $\alpha_{MCR} = M/k$, $\gamma_{MCR} = (n-1)r\beta$ and $\beta = M/[k(n-k)]$, the storage cost is $(M/k)\cdot n$ and the maintenance bandwidth is $[(n-1)/(n-k)]\cdot(M/k)\cdot r$.

MSR: In [10], when using MSR, it is proven that if a new node is allowed to connect to d survival nodes, it needs to store M/k and consume a traffic of $[d/(d-k+1)]\cdot(M/k)$ to repair the data. So the storage cost is $(M/k)\cdot n$, and the maintenance bandwidth is $[d/(d-k+1)]\cdot(M/k)\cdot r$.

MBR: In [10], when using MBR, it is proven that if a new node is allowed to connect to d surviving nodes, it needs to store $[2d/(2d-k+1)]\cdot(M/k)$ and consume a traffic of $[2d/(2d-k+1)]\cdot(M/k)$ to repair the data. So the storage cost is $[2d/(2d-k+1)]\cdot(M/k)\cdot n$, and the maintenance bandwidth is $[2d/(2d-k+1)]\cdot(M/k)\cdot r$.

In order to compare the above redundancy recovery mechanisms while keeping the same redundancy level from multi-loss recovery, we give two factors, a short-term redundancy factor R_s and a long-term redundancy factor R_l, which are introduced in Total Recall prototype [3]. The former is used to tolerate transient failures, i.e., if the available redundancy for a given file falls below R_s, a repair is triggered. The latter is used for each file to accommodate host failures without having to perform frequent file repairs. Correspondingly in this paper, $R_s = (n-r)/k$ and $R_l = n/k$. For example, with $R_s = 2$ and $R_l = 6$ (i.e. $n = 6k$ and $r = 4k$), when $4k$ of $6k$ original nodes become unavailable, the multi-loss recovery is triggered, and all the lost fragments are repaired at $4k$ new nodes.

Moreover, it is proved that [9] a larger d always implies a better or equal storage-repair bandwidth tradeoff of MSR and MBR. So we set d as the maximum $d = n-r$ to compare MCR with the best tradeoff of MSR and MBR.

With R_s, R_l and $d = n-r$, the storage costs and maintenance bandwidths of different redundancy recovery mechanisms are shown in Table I.

Some numerical comparisons are given in Table II and Table III with R_s =2, and R_l = 4 or 8, where R_s and R_l are set the same as in [3]. From Table II, compared with MSR, MCR reduces 22% maintenance bandwidth while keeping the same

TABLE II
$n = 16, r = 8, k = 4$ (R_s =2, R_l = 4)

	Total node storage cost	Maintenance bandwidth
IEC	$4M$	$2M$
MCR	$4M$	$2.5M$
MSR	$4M$	$3.2M$
MBR	$4.9M$	$2.5M$

TABLE III
$n = 32, r = 24, k = 4$ (R_s =2, R_l = 8)

	Total node storage cost	Maintenance bandwidth
IEC	$8M$	$6M$
MCR	$8M$	$6.6M$
MSR	$8M$	$9.6M$
MBR	$9.8M$	$7.4M$

storage cost. And compared with MBR, MCR reduces 23% storage cost while keeping the same maintenance bandwidth. From Table III, compared with MSR, MCR reduces 23% maintenance bandwidth while keeping the same storage cost. And compared with MBR, MCR reduces 23% storage cost and 11% maintenance bandwidth.

So we can conclude that MCR has a better performance in the storage cost and maintenance bandwidth in multi-loss recovery of distributed storage systems compared with other non-cooperative recovery mechanisms.

IV. MCR Transmission and Coding Schemes

Theorem 1 gives a lower bound of maintenance bandwidth with $\beta = M/[k(n-k)]$. In this section, we will construct a recovery transmission scheme in MCR based on $\beta = M/[k(n-k)]$ and a linear coding scheme based on *Strong-MDS code*. So the lower bound in Theorem 1 is tight and the proposed transmission and coding schemes in MCR are optimal with the minimum maintenance bandwidth.

A. Transmission scheme in MCR

To satisfy $\beta = M/[k(n-k)]$, the original file of size M is represented as $k(n-k)$ packets in MCR, each of size $M/[k(n-k)]$. So β equals to the size of one packet. Each fragment of size M/k mentioned before corresponds to $n-k$ packets in MCR. According to the flow analysis of MCR, we specify a recovery transmission scheme of $DSN(n,k,r)$ as follows.

MCR transmission scheme:

Assumptions: The entries of $P_{i,j}, B_{i,j}, B'_{j',j}, Z_{j,i}$ are randomly selected from a finite field F in the following scheme. Suppose that the original file is represented as a matrix $W = (w_{ij})_{k(n-k)\times z}$, where w_{ij} is defined in a finite field F. The file can be viewed as $k(n-k)$ packets, each of size z.

(1) File distribution

1.1) Choose a set of n nodes $X_1, \ldots, X_n$ from idle nodes for a file distribution.

1.2) Encode the $k(n-k)$ packets of the original file into $n(n-k)$ packets $x_{1,1}, \ldots, x_{1,n-k}, \ldots, x_{n,1}, \ldots, x_{n,n-k}$ and $x_{i,j} = P_{i,j}W$, where $P_{i,j}$ is a $k(n-k)$ dimensional coefficient vector, $1 \le i \le n, 1 \le j \le n-k$.

1.3) Each initial node X_i stores $n-k$ packets $x_{i,1}, \ldots, x_{i,n-k}$ as a fragment $(x_{i,1}^T, \ldots, x_{i,n-k}^T)^T = ((P_{i,1}W)^T, \ldots, (P_{i,n-k}W)^T)^T = P_iW$, where $P_i = (P_{i,1}^T, \ldots, P_{i,n-k}^T)^T$ is a $(n-k)\times k(n-k)$ matrix.

(2) Data recovery from r failed nodes ($X_1, \ldots, X_{n-r}$ are still available; $X_{n-r+1}, \ldots, X_n$ become unavailable)

2.1) Choose a set of r new nodes $Y_1, \ldots, Y_r$ from idle nodes for repairing.

2.2) Each old node X_i transmits one encoded packet $\beta_{i,j} = B_{i,j}(x_{i,1}^T, \ldots, x_{i,n-k}^T)^T$ to new node Y_j, where $B_{i,j}$ is a $n-k$ dimensional coefficient vector.

2.3) Each new node $Y_{j'}$ transmits one encoded packet $\beta'_{j',j} = B'_{j',j}(\beta_{1,j'}^T, \ldots, \beta_{n-r,j'}^T)^T$ to each of the other new nodes $Y_j, j \ne j'$, where $B'_{j',j}$ is a $n-r$ dimensional coefficient vector.

2.4) Each new node Y_j encodes $n-1$ accepted packets $\beta_{1,j}, \ldots, \beta_{n-r,j}, \beta'_{1,j}, \ldots, \beta'_{r,j}$ into $n-k$ linearly independent packets $y_{j,1}, \ldots, y_{j,n-k}$, and $y_{j,i} = Z_{j,i}(\beta_{1,j}^T, \ldots, \beta_{n-r,j}^T, \beta_{1,j}^T, \ldots, \beta_{r,j}^T)^T$, where $Z_{j,i}$ is a $n-1$ dimensional coefficient vector. Let $y_{j,i} = Q_{j,i}W$, where $Q_{j,i}$ is a $k(n-k)$ dimensional coefficient vector.

2.5) Each new node Y_j stores $n-k$ packets $y_{j,1}, \ldots, y_{j,n-k}$ as a fragment $(y_{j,1}^T, \ldots, y_{j,n-k}^T)^T = ((Q_{j,1}W)^T, \ldots, (Q_{j,n-k}W)^T)^T = Q_jW$, where $Q_j = (Q_{j,1}^T, \ldots, Q_{j,n-k}^T)^T$ is a $(n-k)\times k(n-k)$ matrix.

B. Coding scheme in MCR

A coding scheme in MCR is valid if a file encoded by this scheme can be reconstructed after multi-loss recovery based on the MCR transmission scheme. In order to obtain the coding scheme, we design an (n,k) strong-MDS code as follows, which is a generalization of MDS code.

(n,k) *Strong-MDS code*: An original file is divided into $k(n-k)$ packets and encoded into $n(n-k)$ packets. The $n(n-k)$ encoded packets are stored at n nodes $X_1, \ldots, X_n$, each node storing $n-k$ encoded packets. If for any of $h_1, h_2, \ldots, h_n$ with $0 \le h_i \le n-k$ and $\sum_{1\le i\le n} h_i = k(n-k)$, each node X_i can selecet h_i encoded packets such that the original file can be reconstructed from the $k(n-k)$ selected packets. We define that this file is (n,k) *strong-MDS* encoded, or, in other words, this file satisfies (n,k) *Strong-MDS Property*.

Our coding scheme in MCR via linear coding is based on the following Lemma 4.

Lemma 4: (Schwartz-Zippel Theorem)[11] Let $P(x_1, \ldots, x_n)$ be a multivariate non-zero polynomial of total degree ρ over a field F. Let S be a finite subset of F and $r_1, \ldots, r_n$ be randomly selected from S. Then $Pr[P(r_1, \ldots, r_n) = 0] \le \frac{\rho}{|S|}$.

Now we need to prove the following Theorem 2 to show that if a file in a distributed storage system is *Strong-MDS* encoded, it will still satisfy *Strong-MDS Property* after multi-loss recovery in MCR.

Theorem 2: Given a distributed storage system with a transmission scheme in MCR and a coding scheme based on (n,k) strong-MDS code. Let an (n,k) *Strong-MDS* encoded file be distributed at $X_1, X_2, \ldots, X_n$. After $X_{n-r+1}, \ldots, X_n$ fail, the lost data are recovered at $Y_1, Y_2, \ldots, Y_r$ based on the MCR transmission scheme. The file stored at $X_1, \ldots, X_{n-r}$ and $Y_1, Y_2, \ldots, Y_r$ will still satisfy (n,k) *Strong-MDS Property* with a probability that can be driven arbitrarily to 1 by increasing the field size of F.

Proof: In the proof of Theorem 2, we use the same assumptions and notations as in the MCR transmission scheme. The coefficient matrix in old node X_i is $P_i = (P_{i,1}^T, \ldots, P_{i,n-k}^T)^T$. There are $n-k$ rows in P_i. Each row is the coefficient vector of a packet in X_i. Similarly, the rows in Q_j are the coefficient vectors of packets in Y_j.

Based on the MCR transmission scheme, the i-th row vector in Q_j packet is

$$Q_{j,i} = Z_{j,i} \times \left(\begin{array}{cc} \begin{pmatrix} B_{1,j} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & B_{n-r,j} \end{pmatrix} & \\ \begin{pmatrix} B'_{1,j} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & B'_{r,j} \end{pmatrix} \cdot & \begin{pmatrix} B_{1,1} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & B_{n-r,1} \\ \ldots & \ldots & \ldots & \ldots \\ B_{1,r} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & B_{n-r,r} \end{pmatrix} \end{array} \right) \times \begin{pmatrix} P_1 \\ \ldots \\ P_{n-r} \end{pmatrix},$$

where $Z_{j,i}$ is of order $1 \times (n-1)$, $B_{i,j}$ is of order $1 \times (n-k)$, $B'_{j',j}$ is of order $1 \times (n-r)$. The entries of $Z_{j,i}$, $B_{i,j}$ and $B'_{j',j}$ are randomly selected from the finite field F.

Because an encoded packet is a linear combination of the packets of the original file, whether the original file can be reconstructed from the $k(n-k)$ selected encoded packets is equivalent to whether the coefficient vectors of the $k(n-k)$ selected encoded packets are mutually linearly independent. So in the following, we concentrate on the analysis of the coefficient matrix and the vectors.

To prove that the file in the storage system still satisfies (n,k) *Strong-MDS Property* after the recovery from multiple losses, it is be equivalent to showing that for any $h_1^{old}, h_2^{old}, \ldots, h_{n-r}^{old}, h_1^{new}, h_2^{new}, \ldots, h_r^{new} (0 \le h_i^{old}, h_j^{new} \le n-k)$ with $\sum_{1 \le i \le n-r} h_i^{old} + \sum_{1 \le j \le r} h_j^{new} = k(n-k)$, there exist h_i^{old} row vectors (corresponding to h_i^{old} packets at X_i, denoted as a matrix E_i^{old} in the following) in P_i and h_j^{new} row vectors (corresponding to h_j^{new} packets at Y_j, denoted as a matrix E_j^{new} in the following) in Q_j, such that

$$Det([E_1^{old\ T}, \ldots, E_{n-r}^{old\ T}, E_1^{new\ T}, \ldots, E_r^{new\ T}]^T) \neq 0 \quad (10)$$

Because each row vector in E_i^{old} is a row vector in P_i and each row vector in E_j^{new} is a row vector in Q_j, $Det([E_1^{old\ T}, \ldots, E_{n-r}^{old\ T}, E_1^{new\ T}, \ldots, E_r^{new\ T}]^T)$ is a multivariate polynomial of total degree ρ, where variables are the entries of $Z_{i,j}, B_{i,j}, B'_{j',j}$.[1]

[1]Note that P_i is the coefficient matrix of the encoded packets stored at X_i. So the entries of P_i are constants.

By Lemma 4, if the left of (10) is a non-zero polynomial, the probability that (10) holds can be driven arbitrarily to 1 by increasing the field size of F.

We will use the following Lemma 5 to show that there is an assignment of $Z_{i,j}, B_{i,j}$ and $B'_{j',j}$ such that the left of (10) is a non-zero polynomial. By Lemma 4 and Lemma 5, Theorem 2 concludes. ∎

In order to prove Lemma 5, we first introduce the following definitions.

Definition 1. Given $h_1^{old}, \ldots, h_{n-r}^{old}, h_1^{new}, \ldots, h_r^{new}$ with $\sum_{1 \le i \le n-r} h_i^{old} + \sum_{1 \le j \le r} h_j^{new} = k(n-k)$, define that $\mathbf{H^{old}} = (h_1^{old}, h_2^{old}, \ldots, h_{n-r}^{old})$, $\mathbf{H^{new}} = (h_1^{new}, h_2^{new}, \ldots, h_r^{new})$, and let E_i^{old} be the matrix of the first h_i^{old} row vectors of P_i, F_i^{old} be the matrix of the $n-k-h_i^{old}$ remaining row vectors of P_i, and E_j^{new} be the matrix of the first h_j^{new} row vectors of Q_j.

Definition 2. Let $h^{old} = \sum_{i=1}^{n-r} h_i^{old}$, $h^{new} = \sum_{j=1}^{r} h_j^{new}$. So $h^{old} + h^{new} = k(n-k)$. If there exist h^{new} row vectors in $F_1^{old}, F_2^{old}, \ldots, F_{n-r}^{old}$, such that these h^{new} row vectors in $F_1^{old}, F_2^{old}, \ldots, F_{n-r}^{old}$ can be transmitted to $Y_1, \ldots, Y_r$ for the construction of h^{new} row vectors of $E_1^{new}, \ldots, E_r^{new}$ with an assignment of $Z_{i,j}, B_{i,j}, B'_{j',j}$, $\mathbf{H^{old}}$ and $\mathbf{H^{new}}$ are called *transmissive*.

Because any $k(n-k)$ vectors stored at $X_1, X_2, \ldots, X_{n-r}$ are linearly independent, whether there exists an assignment of $Z_{i,j}, B_{i,j}, B'_{j',j}$ such that the left of (10) is non-zero polynomial is equivalent to whether $\mathbf{H^{old}}$ and $\mathbf{H^{new}}$ are *transmissive* for any $h_1^{old}, h_2^{old}, \ldots, h_{n-r}^{old}, h_1^{new}, h_2^{new}, \ldots, h_r^{new} (0 \le h_i^{old}, h_j^{new} \le n-k)$ with $\sum_{1 \le i \le n-r} h_i^{old} + \sum_{1 \le j \le r} h_j^{new} = k(n-k)$. We will use the following Lemma 5 to show that $\mathbf{H^{old}}$ and $\mathbf{H^{new}}$ are *transmissive*.

Lemma 5: $\mathbf{H^{old}}$ and $\mathbf{H^{new}}$ are *transmissive* for any $h_1^{old}, h_2^{old}, \ldots, h_{n-r}^{old}, h_1^{new}, h_2^{new}, \ldots, h_r^{new} (0 \le h_i^{old}, h_j^{new} \le n-k)$ with $\sum_{1 \le i \le n-r} h_i^{old} + \sum_{1 \le j \le r} h_j^{new} = k(n-k)$.

To prove Lemma 5, we need the following Proposition 1 to 3.

Proposition 1: $\mathbf{H^{old}} = (0, 0, \ldots, 0, \ldots, c^{old}, n-k, n-k, n-k)$ and $\mathbf{H^{new}} = (n-k, n-k, \ldots, n-k, c^{new}, 0, 0, \ldots, 0)$ are *transmissive*, where $c^{old} = h^{old} - (n-k)\lfloor \frac{h^{old}}{n-k} \rfloor$, $c^{new} = h^{new} - (n-k)\lfloor \frac{h^{new}}{n-k} \rfloor$, the number of element $n-k$ in $\mathbf{H^{old}}$ and the number of element $n-k$ in $\mathbf{H^{new}}$ are $h^{old} mod (n-k)$ and $h^{new} mod (n-k)$ respectively.

Proof: Let the number of element 0 in $\mathbf{H^{old}}$ be σ. So there are $n-r-\sigma-1$ element $n-k$ in $\mathbf{H^{old}}$, and one element c^{old} (may be 0) in $\mathbf{H^{old}}$.

Let the number of element $n-k$ in $\mathbf{H^{new}}$ be τ. So there are $r-\tau-1$ element 0 and one element c^{new} (may be 0) in $\mathbf{H^{new}}$.

Because $h^{old} + h^{new} = k(n-k)$,

$$(n-k)(n-r-\sigma-1) + c^{old} + (n-k)\tau + c^{new} = k(n-k). \quad (11)$$

From the assumptions in Proposition 1,

$$\begin{aligned} c^{old}+c^{new} &= h^{old}-(n-k)\lfloor\frac{h^{old}}{n-k}\rfloor \\ &\quad +h^{new}-(n-k)\lfloor\frac{h^{new}}{n-k}\rfloor \\ &= n-k. \end{aligned} \tag{12}$$

From (11) and (12),

$$\sigma-\tau=n-r-k \tag{13}$$

and

$$\sigma+r-\tau=n-k \tag{14}$$

hold. With $n-k\geq r$, (13) implies

$$\sigma\geq\tau. \tag{15}$$

Let the number of selected vectors in $F_1^{old},F_2^{old},\ldots,F_{n-r}^{old}$ transmitted to Y_j be s_i and $\mathbf{S}=(s_1,s_2,\ldots,s_r)$ specifies the numbers of row vectors transmitted to $Y_1,Y_2,\ldots,Y_r$ during the process of the MCR transmission scheme.

Initially in the MCR transmission scheme, $\mathbf{S}=(0,0,\ldots,0)$.

From 2.2) of the MCR transmission scheme, because $n-k\geq r,r$ different row vectors of F_i^{old} containing $n-k$ row vectors can be transmitted to $Y_1,Y_2,\ldots,Y_r$ by an assignment of $B_{i,j}$, where each of $Y_1,Y_2,\ldots,Y_r$ receives a vector. The number of F_i^{old} containing $n-k$ row vectors is σ, so the number of selected vectors transmitted from F_i^{old} to each of $Y_1,Y_2,\ldots,Y_r$ is σ, and thus $\mathbf{S}=(\sigma,\sigma,\ldots,\sigma)$ after Step 2.2) in MCR.

From 2.3) of the MCR transmission scheme, because (15), $Y_j(\tau+1\leq j\leq r)$ can transmit τ different vectors of s_j accepted vectors to $Y_1,Y_2,\ldots,Y_\tau$ by an assignment of $B'_{j',j}$, each of $Y_1,Y_2,\ldots,Y_\tau$ receives a vector. So $\mathbf{S}=(\sigma+r-\tau,\ldots,\sigma+r-\tau,\sigma-\tau,\ldots,\sigma-\tau)$ after the transmission of these vectors, where the number of element $\sigma+r-\tau$ is τ . Because (14), $\mathbf{S}=(n-k,n-k,\ldots,n-k,\sigma-\tau,\ldots,\sigma-\tau)$, where the number of element $n-k$ is τ.

Let $i_0=n-r-h^{old}mod(n-k)$. There are $l_i^{old}=n-k-c^{old}$ row rectors in $F_{i_0}^{old}$. X_{i_0} transmits l_i^{old} row vectors to $Y_1,Y_2,\ldots,Y_{n-k-c^{old}}$, each of which receives a different vector. And then $Y_1,Y_2,\ldots,Y_{n-k-c^{old}}$ can transmit these vectors to $Y_{\tau+1}$ by an assignment of $B_{i,j}$ and $B'_{j',j}$ in Step 2.3). So $\mathbf{S}=(n-k,n-k,\ldots,n-k,\sigma-\tau+n-k-c^{old},\sigma-\tau,\ldots,\sigma-\tau)$ after Step 2.3).

From the above analysis, some row vectors in $F_1^{old},F_2^{old},\ldots,F_{n-r}^{old}$ can be transmitted to $Y_1,Y_2,\ldots,Y_r$, where $Y_1,Y_2,\ldots,Y_r$ receive $n-k,n-k,\ldots,n-k,\sigma-\tau+n-k-c^{old},\sigma-\tau,\ldots,\sigma-\tau$ vectors respectively.

From (12) and (15), $\sigma-\tau+n-k-c^{old}\geq c^{new}$, and $\sigma-\tau\geq 0$. So we can set $Z_{i,j}$ to reduce $(n-k,n-k,\ldots,n-k,\sigma-\tau+n-k-c^{old},\sigma-\tau,\ldots,\sigma-\tau)$ to $(n-k,n-k,\ldots,n-k,c^{new},0,\ldots,0)$.

Therefore, $\mathbf{H^{old}}=(0,0,\ldots,0,\ldots,c^{old},n-k,c^{old},n-k)$ and $\mathbf{H^{new}}=(n-k,n-k,\ldots,n-k,c^{new},0,\ldots,0)$ are *transmissive*.

Proposition 1 concludes. ∎

Proposition 2. If $\mathbf{H^{old}}=(h_1^{old},h_2^{old},\ldots,h_{n-r}^{old})$ and $\mathbf{H^{new}}=(h_1^{new},h_2^{new},\ldots,h_r^{new})$ are *transmissive* with $h_i^{old}<h_{i'}^{old}$ for some $1\leq i\neq i'\leq n-r$, $\mathbf{H^{old}}=(h_1^{old},\ldots,h_i^{old}+1,\ldots,h_{i'}^{old}-1,\ldots,h_{n-r}^{old})$ and $\mathbf{H^{new}}=(h_1^{new},h_2^{new},\ldots,h_r^{new})$ are also *transmissive*.

Proof: The proof of Proposition 2 is similar to but easier than Proposition 3. So omitted. ∎

Proposition 3. If $\mathbf{H^{old}}=(h_1^{old},h_2^{old},\ldots,h_{n-r}^{old})$ and $\mathbf{H^{new}}=(h_1^{new},h_2^{new},\ldots,h_r^{new})$ are transmissive with $h_j^{new}>h_{j'}^{new}$ for some $1\leq j\neq j'\leq r$, $\mathbf{H^{old}}=(h_1^{old},h_2^{old},\ldots,h_{n-r}^{old})$ and $\mathbf{H^{new}}=(h_1^{new},\ldots,h_j^{new}-1,\ldots,h_{j'}^{new}+1,,h_r^{new})$ are also *transmissive*.

Proof: Because $Y_1,Y_2,\ldots,Y_r$ are order independent, we assume that $h_1^{new},h_2^{new},\ldots,h_r^{new}$ are in descending order without loss of generality. Because $h_1^{new},h_2^{new},\ldots,h_r^{new}$ are in descending order, it is enough to consider any $h_j^{new}>h_{j'}^{new}$ with $j<j'$. Since $h_j^{new}>h_{j'}^{new}$, there exists X_i which transmits a row vector to Y_j but does not transmit a row vector to $Y_{j'}$ or there exists $Y_{j''}$ which transmits a row vector to Y_j but does not transmit a row vector to $Y_{j'}$. In the above two situations, we can adjust the assignment of $B_{i,j},B'_{j',j}$ such that X_i transmits the row vector to $Y_{j'}$ but not to Y_j , or $Y_{j''}$ transmits the row vector to $Y_{j'}$ but not to Y_j.

Proposition 3 concludes. ∎

Proof of Lemma 5: From Proposition 2 and Proposition 3, $(0,0,\ldots,0,\ldots,c^{old},n-k,n-k,n-k)$ and $(n-k,n-k,\ldots,n-k,c^{new},0,\ldots,0,\ldots,0)$ can be transferred into any $\mathbf{H^{old}}$ and $\mathbf{H^{new}}$ while keeping transmissive. From Proposition 1, $(0,0,\ldots,0,\ldots,c^{old},n-k,n-k,n-k)$ and $(n-k,n-k,\ldots,n-k,c^{new},0,0,\ldots,0)$ are *transmissive*. So any $\mathbf{H^{old}}$ and $\mathbf{H^{new}}$ are transmissive.

So Lemma 5 concludes. ∎

From the above analysis, Theorem 2 holds. So a file in a distributed storage system with MCR transmission scheme will keep (n,k) strong-MDS property with a probability that can be driven arbitrarily to 1 by increasing the field size of F. So the original file can be reconstructed after multiple losses with a probability that can be driven arbitrarily to 1 by increasing the field size of F. Thus our recovery transmission scheme and coding scheme in MCR based on $\beta=M/[k(n-k)]$ are valid.

V. Conclusion

This paper gave the tight lower bound of maintenance bandwidth for the cooperative recovery from multiple losses in distributed storage system and also designed a MCR transmission scheme matching the tight lower bound. To design a MCR coding scheme, we presented a *strong-MDS* code and showed the existence of a random linear *strong-MDS* code with a sufficient large finite field. The decoding complexity of our random linear coding scheme is expensive, and our future work is to design a determinate coding algorithm to reduce the decoding complexity.

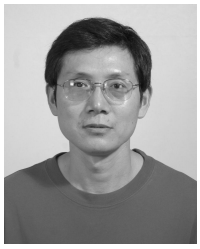
References

[1] Antony Rowstron and Peter Druschel, "Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility," in *Proc. ACM SOSP'01*, Oct. 2001.

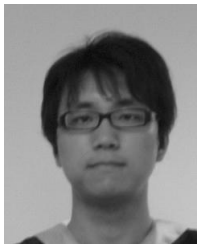
[2] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS," in *Proc. ACM SOSP'01*, Oct. 2001.

[3] R. Bhagwan, K. Tati, Y. Cheng, S. Savage, and G. Voelker, "Total recall: System support for automated availability management," in *Proc. NSDI'01*, Mar. 2004.
[4] Atul Adya, William J. Bolosky, Miguel Castro, Ronnie Chaiken, Gerald Cermak, John R. Douceur, John Howell, Jacob R. Lorch, Marvin Theimer, and Roger Wattenhofer. "FARSITE: Federated, available, and reliable storage for an incompletely trusted environment," in *Proc. OSDI'05*, Dec. 2002.
[5] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. "OceanStore: An architecture for global-scale persistent storage," in *Proc. ASPLOS 2000*, Dec. 2000.
[6] Rodrigues, R., and Liskov, B. "High availability in DHTs: Erasure coding vs. replication," in *Proc. 4th International Workshop on Peer-to-Peer Systems*, Feb. 2005.
[7] M. Rabin. "Efficient dispersal of information for security, load balancing, and fault tolerance," in *J. ACM 36, 2* (1989), 335-348.
[8] A. G. Dimakis, P. B. Godfrey, M. Wainwright, and K. Ramchandran. "Network Coding for Distributed Storage Systems," in *Proc. IEEE INFOCOM 2007*, May 2007.
[9] Y. Wu, A. G. Dimakis, and K. Ramchandran. "Deterministic regenerating codes for distributed storage," in *Allerton Conference on Control, Computing, and Communication,* (Urbana-Champaign, IL), Sep. 2007.
[10] Alexandros G. Dimakis, P. Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, Kannan Ramchandran. "Network Coding for Distributed Storage Systems," in *arXiv:0803.0632v1 [cs.NI]*, Mar. 2008.
[11] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.
[12] Weatherspoon, H., and Kubiatowicz J. D. "Erasure coding vs. replication: A quantitative comparison," in *Proc. 1st IPTPS*, Mar. 2002.

YuChong Hu received the B.S. from the School for the Gifted Young, University of Science & Technology of China, Anhui, China, in 2005. He is currently working toward the Ph.D. degree at the School of Computer Science, University of Science & Technology of China. His research interests include network coding, distributed storage and cloud storage.



Yinlong Xu received his B.S. in Mathematics from Peking University in 1983, and MS and Ph.D in Computer Science from University of Science & Technology of China(USTC) in 1989 and 2004 respectively. He is currently a professor with the School of Computer Science & Technology at USTC. Prior to that, he served the Department of Computer Science & Technology at USTC as an assistant professor, a lecturer, and an associate professor. Currently, he is leading a group of research students in doing some networking and high performance computing research. His research interests include network coding, wireless network, combinatorial optimization, design and analysis of parallel algorithm, parallel programming tools, etc. He received the Excellent Ph.D Advisor Award of Chinese Academy of Sciences in 2006.



XiaoZhao Wang received the B.S. from the School of Computer Science, University of Science & Technology of China, Anhui, China, in 2009. His research interests include network coding and the ranking of web page. He now works as a research engineer in the company of Baidu(Japan).

Cheng Zhan received the B.S. from the School of Computer Science, University of Science & Technology of China, Anhui, China, in 2006. He is currently working toward the Ph.D. degree at the School of Computer Science, University of Science & Technology of China. His research interests include network coding, reliable multicast and on demand broadcast.

Pei Li received the B.S. and Ph.D both from the School of Computer Science, University of Science & Technology of China, Anhui, China, in 2004 and 2009 respectively. His research interests include wireless network and peer-to-peer network.